

BAB II

LANDASAN TEORI

2.1. Decision Support System

Decision Support System atau Sistem pendukung keputusan adalah program terkomputerisasi yang digunakan untuk mendukung penentuan, penilaian, dan kursus tindakan dalam organisasi atau bisnis. Sistem ini menyaring dan menganalisis data dalam jumlah besar, mengumpulkan informasi komprehensif yang dapat digunakan untuk memecahkan masalah dan dalam pengambilan keputusan [6]. Istilah Sistem Pendukung Keputusan pertama kali digagas oleh P.G.W Keen, seorang akademisi Inggris yang kemudian melanjutkan karir di USA. Pada tahun 1978 Keen dan Scott Morton menerbitkan sebuah buku dengan judul *Decision Support Systems : An Organisation Perspective*. Dimana dalam buku tersebut mereka menyebutkan bahwa sistem komputer berdampak pada keputusan yang akan dibuat, karena komputer dan analisis merupakan faktor penting untuk dipertimbangkan dalam menetapkan sebuah keputusan [7]. Sistem pendukung keputusan dirancang untuk menunjang seluruh tahapan pembuatan keputusan yang dimulai dari tahap identifikasi masalah, pemilihan data yang relevan, menentukan pendekatan yang digunakan dalam proses pembuatan keputusan, sampai pada kegiatan mengevaluasi pemilihan alternatif.

2.2. Database System

Database adalah kumpulan data yang saling terkait yang disimpan bersama dengan redundansi terkontrol untuk melayani satu atau lebih aplikasi secara optimal. Data tersebut disimpan sedemikian rupa bahwa mereka tidak bergantung pada program yang digunakan oleh masyarakat untuk mengakses data. Itu pendekatan yang digunakan dalam menambahkan data baru, memodifikasi dan mengambil data yang ada dari database umum dan terkontrol. Ini juga didefinisikan sebagai kumpulan data terkait secara logis yang disimpan bersama yang dirancang untuk memenuhi kebutuhan informasi suatu organisasi. Kita juga bisa mendefinisikannya sebagai sistem pengisian elektronik [8].

2.3. PHPMyAdmin

PHPMyAdmin adalah sebuah aplikasi yang menyediakan *interface website* yang lengkap untuk administrasi *database MySQL*. Tujuan utama dari *PHPMyAdmin* itu sendiri adalah untuk memberikan *web-based management* untuk *server* ataupun *database* yang berbasis *MySQL*. Di dalam *PHPMyAdmin* ini kita bisa membuat ataupun meng-*import database* baru, membuat tabel, serta membuat relasi antar tabel [9].

2.4. Multiple Attribute Decision Making

Multiple Attribute Decision Making (MADM) adalah suatu metode yang digunakan untuk mencari alternatif optimal dengan mengambil banyak kriteria sebagai dasar pengambilan keputusan, dengan penilaian yang subjektif dimana analisis matematis tidak terlalu banyak digunakan untuk pemilihan alternatif dalam jumlah yang sedikit [10].

2.5. Simple Additive Weighting (SAW)

Simple Additive Weighting atau SAW adalah suatu metode *decision support system* yang bisa dibilang paling sering digunakan karena kemudahannya. Konsep dasar dari SAW adalah pencarian jumlah bobot dari *performance rating* semua atribut. [11]. Metode ini memiliki beberapa rumus untuk mencari nilai-nilai yang dibutuhkan, seperti rumus normalisasi dan rumus *value of preferences*.

$$R_{ij} \left\{ \frac{x_{ij}}{\max_i x_{ij}} \right.$$

Rumus 2. 1. Rumus Normalisasi (Benefit)

$$R_{ij} \left\{ \frac{x_{ij}}{\min_i x_{ij}} \right.$$

Rumus 2. 2. Rumus Normalisasi (Cost)

Rumus diatas merupakan rumus dari normalisasi, dimana penjelasan dari rumus tersebut adalah sebagai berikut:

- R_{ij} : Hasil dari Normalisasi.
- \max_i : Nilai maksimum dari setiap baris dan kolom.
- \min_i : Nilai minimum dari setiap baris dan kolom.
- x_{ij} : Baris dan kolom dari matrix.

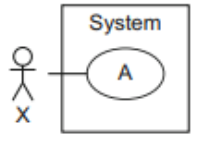

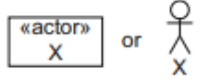
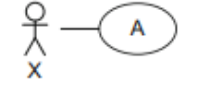
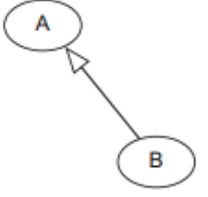
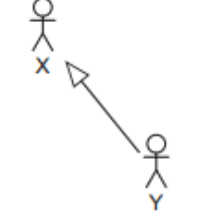
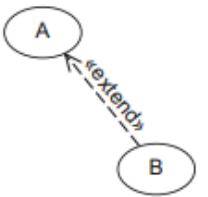
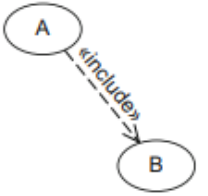
$$V_i = \sum_{j=1}^n W_j P_{ij}$$

Rumus 2. 3. Rumus Value of Preferences

Gambar diatas merupakan rumus dari pencarian nilai *value of preferences*, dimana penjelasan dari rumus tersebut adalah sebagai berikut:

- a. V_i : Hasil akhir nilai preferensi.
- b. P : Elemen baris matriks ternormalisasi.
- c. W : Bobot Preferensi

2.6. Use Case Diagram

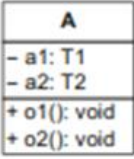
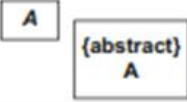
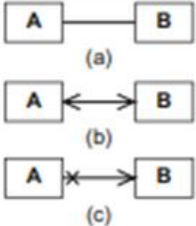
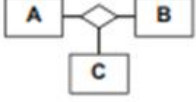
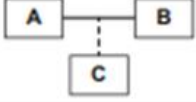
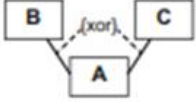

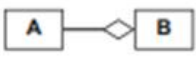



<i>Name</i>	<i>Notation</i>	<i>Description</i>
System		Boundaries between the system and the users of the system
Use case		Unit of functionality of the system
Actor		Role of the users of the system
Association		X participates in the execution of A
Generalization (use case)		B inherits all properties and the entire behavior of A
Generalization (actor)		Y inherits from X; Y participates in all use cases in which X participates
Extend relationship		B extends A: optional incorporation of use case B into use case A
Include relationship		A includes B: required incorporation of use case B into use case A

Gambar 2. 1. Notasi Use Case Diagram

Sumber: [12]

Gambar 2.3 di atas adalah penjelasan tentang berbagai notasi yang ada pada *use case* diagram. *Use case diagram* itu sendiri merupakan diagram yang membantu mendeskripsikan berbagai skenario dalam penggunaan suatu sistem. Diagram tersebut membantu mengekspresikan berbagai hal yang sebuah sistem harus lakukan tanpa penjelasan mengenai *data structures*, ataupun algoritma [12].

2.7. Class Diagram






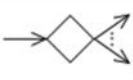

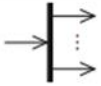
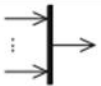

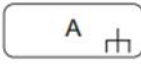



Name	Notation	Description
Class		Description of the structure and behavior of a set of objects
Abstract class		Class that cannot be instantiated
Association		Relationship between classes: navigability unspecified (a), navigable in both directions (b), not navigable in one direction (c)
N-ary association		Relationship between <i>N</i> (in this case 3) classes
Association class		More detailed description of an association
xor relationship		An object of <i>A</i> is in a relationship with an object of <i>B</i> or with an object of <i>C</i> but not with both
Strong aggregation = composition		Existence-dependent parts-whole relationship (<i>A</i> is part of <i>B</i> ; if <i>B</i> is deleted, related instances of <i>A</i> are also deleted)
Shared aggregation		Parts-whole relationship (<i>A</i> is part of <i>B</i> ; if <i>B</i> is deleted, related instances of <i>A</i> need not be deleted)
Generalization		Inheritance relationship (<i>A</i> inherits from <i>B</i>)
Object		Instance of a class
Link		Relationship between objects

Gambar 2. 2. Notasi Class Diagram

Sumber: [12]

Gambar 2.2 diatas menjelaskan notasi-notasi yang ada dalam pembuatan *class diagram*. *Class diagram* itu sendiri digunakan untuk membuat model struktur sebuah sistem, dimana diagram tersebut menjelaskan mengenai berbagai elemen dalam sebuah sistem dan relasi antara elemen-elemen tersebut.

2.8. Activity Diagram

Name	Notation	Description
Action node		Actions are atomic, i.e., they cannot be broken down further
Activity node		Activities can be broken down further
Initial node		Start of the execution of an activity
Activity final node		End of ALL execution paths of an activity
Flow final node		End of ONE execution path of an activity
Decision node		Splitting of one execution path into two or more alternative execution paths
Merge node		Merging of two or more alternative execution paths into one execution path
Parallelization node		Splitting of one execution path into two or more concurrent execution paths
Synchronization node		Merging of two or more concurrent execution paths into one execution path
Edge		Connection between the nodes of an activity
Call behavior action		Action A refers to an activity of the same name
Object node		Contains data and objects that are created, changed, and read
Parameters for activities		Contain data and objects as input and output parameters
Parameters for actions (pins)		Contain data and objects as input and output parameters

Gambar 2. 3. Notasi Activity Diagram

Sumber: [12]

Gambar 2.3 diatas menjelaskan mengenai notasi-notasi yang ada dalam pembuatan *activity* diagram. Dalam perancangan sistem, *activity* diagram digunakan untuk menggambarkan berbagai aktivitas, objek, event dan state dalam sebuah sistem. Tujuan dari pembuatan diagram tersebut adalah menggambarkan perilaku sebuah sistem untuk berbagai aktivitas [13].

2.9. MySQL

MySQL adalah salah satu *RDBMS* atau *Relational Database Management System* yang cukup terkenal dan banyak digunakan. *MySQL* memungkinkan user untuk membuat, menyimpan, menyusun, dan mendapatkan data dengan mudah. *MySQL* juga mengontrol akses dan memastikan pengguna atau user dapat mengakses database secara bersamaan. *Server MySQL* menggunakan Bahasa pemrograman *SQL (Standard Query Language)*. Bahasa pemrograman tersebut juga cukup populer dan digunakan pada penyedia *RDBMS* lainnya seperti Oracle, Microsoft SQL Server, dll. [14].

2.10. HTML (Hypertext Markup Language)

HTML atau *Hypertext Markup Language* adalah bahasa komputer yang dirancang untuk memungkinkan pembuatan situs web. Situs web ini kemudian dapat dilihat oleh siapa pun yang terhubung ke *Internet*. HTML juga relatif mudah dipelajari. HTML terdiri dari serangkaian kode pendek yang diketik ke dalam file teks oleh penulis situs - ini adalah tag. Teks tersebut kemudian disimpan sebagai *file html*, dan dilihat melalui browser, seperti Internet Explorer atau Netscape Navigator. *Browser* membaca file dan menerjemahkan teks ke dalam bentuk yang terlihat, semoga menampilkan halaman seperti yang diinginkan penulis. Menulis

HTML Anda sendiri memerlukan penggunaan tag dengan benar untuk menciptakan visi Anda. [15].

2.11. PHP

PHP adalah bahasa pemrograman sisi server yang dirancang khusus untuk web. Di dalam halaman HTML, Anda dapat menanamkan kode PHP yang akan dijalankan setiap kali halaman tersebut dikunjungi. Kode PHP Anda adalah ditafsirkan di *server web* dan menghasilkan HTML atau keluaran lain yang akan dilihat pengunjung. *Tags* php dimulai dengan “?php” dan diakhiri dengan “?>”. *Tags* ini biasanya berisi *script* atau kode yang terhubung ke *server-side website* ataupun *database*. PHP bersifat *open source*, yang berarti Anda memiliki akses ke kode sumber dan memiliki ekstensi kebebasan untuk menggunakan, mengubah, dan mendistribusikannya kembali. Bisa dibilang bahwa PHP ini adalah jembatan atau penghubung antara halaman depan suatu *web* dan *database* [14].

2.12. Metode SDLC Waterfall

Metode *SDLC (System Development Life Cycle) Waterfall* merupakan model pengembangan sistem yang dilakukan secara berurutan tanpa tumpang tindih, mulai dari *Requirement, Analysis, Design System, Coding & Implementation, Testing, Operation/Deployment*, hingga tahap terakhir yaitu *maintenance system*. Tahap *Requirement* adalah tahapan pengumpulan data atau informasi yang dibutuhkan oleh sistem seperti fitur yang akan ditambahkan ke dalam sistem. Tahap *Design System* adalah tahapan dimana gambaran sistem akan jadi seperti apa sistem ini nantinya dibuat. Tahap *Coding & Implementation* adalah tahap penulisan *script* yang akan ditanamkan ke sistem. Tahapan *Testing*

merupakan tahapan dimana *user* melakukan percobaan terhadap sistem yang telah dibuat, untuk mencari *bug* ataupun *error* yang kemungkinan terjadi kepada sistem. Selanjutnya tahapan *Operation/Deployment* dimana sistem yang telah dites diluncurkan untuk digunakan oleh *user*. Terakhir yaitu tahap *Maintenance* dimana perawatan sistem akan dilakukan secara berkala setelah sistem dirilis dan dipublikasikan [16].

2.13. Valorant

Valorant merupakan *video game multiplayer first-person shooter* buatan Riot Games, *developer video games* dari Amerika Serikat. Valorant sendiri dirilis pada tanggal 2 Juni 2020 dan langsung memikat banyak *gamers* di seluruh dunia dikarenakan konsepnya yang unik, banyak pemain yang berkata bahwa Valorant ini seperti menggabungkan konsep permainan CS:GO dan Overwatch yang telah rilis sebelumnya [5]. Di dalam permainan ini terdapat *agents* atau karakter yang dapat dipilih oleh para pemain. Para *agents* ini memiliki *skill* atau kemampuan yang unik dan berbeda-beda, sehingga penggunaan pada permainan nya pun juga beragam. Permainan ini melibatkan 10 pemain dalam satu pertandingan, yang dimana masing-masing tim terdiri dari 5 pemain dimana masing-masing tim berusaha memenangkan pertandingan yang terdiri dari total 25 ronde. Salah satu tim bisa dikatakan menang jika sudah meraih 13 poin, atau tim lawan menyatakan menyerah.

2.14. Video Games

Video Games adalah suatu sarana *entertainment* yang bisa “dimainkan” oleh pemain menggunakan komputer, konsol *video games*, telfon genggam,

ataupun tablet. *Video Games* pertama kali dibuat dan diperkenalkan pada tahun 1958, oleh seorang fisikawan bernama William Higinbotham. William Higinbotham membuat *game* tennis simpel yang melibatkan dua pemain di dalamnya. Setelah itu *video games* berkembang sangat pesat hingga banyak sekali peminatnya seperti sekarang [17].

2.15. Penelitian Terdahulu

Tabel 2. 1 Penelitian Terdahulu 1

No.	Jurnal	Keterangan
1.	Penulis	Sri Wulandari, Adityo Permana Wibowo
	Nama Jurnal	International Journal of Engineering, Technology and Natural Sciences ISSN 2685-3191 Vol. 1, No. 1 Juli 2019
	Judul Jurnal	DEVELOPMENT OF SAW (SIMPLE ADDITIVE WEIGHTING) METHOD FOR DECISION SUPPORT SYSTEM OF SEMBAKO PRICE CONTROL
	Metode	Metode SPK: Simple Additive Weighting (SAW) Metode pengumpulan data : Observasi dan survei
	Hasil	Pembuatan Aplikasi Sistem Pendukung Keputusan ini dapat dengan cepat dan efisien melakukan proses penyelesaian masalah pengendalian harga sembako yang semakin meningkat, sehingga informasi pendukung untuk solusi yang diperoleh lebih cepat.

Tabel 2. 2 Penelitian Terdahulu 2

No.	Jurnal	Keterangan
2.	Penulis	Febri Haswan
	Nama Jurnal	International Journal Of Artificial Intelligence Research
	Judul Jurnal	Decision Support System For Election Of Members Unit Patients Pamong Praja
	Metode	Metode SPK: Simple Additive Weighting (SAW) Metode pengumpulan data : Observasi dan Studi Literatur
	Hasil	Dengan adanya sistem pendukung keputusan dalam pemilihan anggota Satpol PP baru dengan menggunakan <i>Simple Additive Weighting Method (SAW)</i> dapat membantu atau sebagai acuan bagi

		pengguna dalam menentukan pemilihan anggota baru Satpol PP.
--	--	---

Tabel 2. 3 Penelitian Terdahulu 3

No.	Jurnal	Keterangan
3.	Penulis	Suwarti
	Nama Jurnal	Jaringan Sistem Informasi Robotik Vol. 2, No. 01, Maret 2018
	Judul Jurnal	Metode Simple Additive Weighting (Saw) Dalam Menentukan Kualitas Tepung Dalam Pembuatan Cup Cake
	Metode	Metode SPK: Simple Additive Weighting (SAW) Metode pengumpulan data : Observasi
	Hasil	Dengan adanya sistem pendukung keputusan dengan metode SAW ini dapat membantu sekaligus memberikan referensi bagi pengguna dalam memilih kualitas tepung untuk pembuatan <i>cup cake</i> .

Tabel 2. 4 Penelitian Terdahulu 4

No.	Jurnal	Keterangan
4.	Penulis	Bambang Efiriyanto
	Nama Jurnal	Electronic Theses and Dissertations Universitas Muhamadiyah Surakarta
	Judul Jurnal	Sistem Pendukung Keputusan Menggunakan Metode <i>Simple Additive Weighting (SAW)</i> untuk Menentukan Karyawan Terbaik pada Dealer Motor Berbasis Web
	Metode	Metode SPK: Simple Additive Weighting (SAW) Metode pengumpulan data : Observasi dan wawancara
	Hasil	Sistem pendukung keputusan untuk menentukan karyawan terbaik pada Dealer Motor ini dirancang untuk menghitung bobot dari setiap alternatif dan kriteria yang telah ditentukan.

Pada penelitian terdahulu yang telah dijelaskan pada tabel 2.1, 2.2, 2.3, dan 2.4 diatas, terdapat berbagai ilmu yang bisa dipelajari dan digunakan untuk membantu jalannya penelitian yang sedang dilakukan Penulis sekarang ini. Contohnya Di semua penelitian terdahulu dapat dilihat metode pengambilan data yang sangat cocok, yaitu metode observasi, dan khususnya pada jurnal pertama yang ditulis oleh Sri Wulandari dan Adityo PermanaWibowo juga dilakukan

dengan metode survei yang menambah kecocokan terhadap penelitian ini. Metode sistem pendukung keputusan yang digunakan oleh jurnal-jurnal diatas juga menggunakan metode *Simple Additive Weighting* sehingga sangat cocok untuk dijadikan acuan dan pembelajaran untuk penelitian ini.